

ROUTING AND TRANSMITTAL SLIP		Date
		23 Mar 83
TO: (Name, office symbol, room number, building, Agency/Post)		Initials
1. EXO/ODP		
2. <u>file - presentations</u>		
3.		
4.		
5.		
Action	File	Note and Return
Approval	For Clearance	Per Conversation
As Requested	For Correction	Prepare Reply
Circulate	For Your Information	See Me
Comment	Investigate	Signature
Coordination	Justify	
REMARKS		

George:

Attached is a copy of the Presentation  
Richard R. Schieffelin (QSI Contractor)  
is presenting in Chicago on 25 March 1983  
at the DPMA Conference.

Thanks 

DO NOT use this form as a RECORD of approvals, concurrences, disposals,  
clearances, and similar actions

FROM: (Name, org. symbol, Agency/Post)	Room No. <small>Block</small>
DC/QAD	2E21
	Phone No.
5041-102	

\* GPO : 1981 O - 341-529 (120)

OPTIONAL FORM 41 (Rev. 7-76)  
Prescribed by GSA  
FPMR (41 CFR) 101-11.206

**QUALITY SYSTEMS INCORPORATED**

8401 ARLINGTON BLVD., FAIRFAX, VIRGINIA 22031

(703) 573-7440

**RICHARD R. SCHIEFFELIN**  
PRINCIPAL QUALITY ASSURANCE ANALYST



**Quality  
Systems  
Incorporated**



# AGENDA

## Experiences in Incorporating QA into a DP Organization

- I. QSI Overview
- II. Relevant Experience
  - A. Steps to consider
  - B. Tools
- III. Summary

**Page Denied**



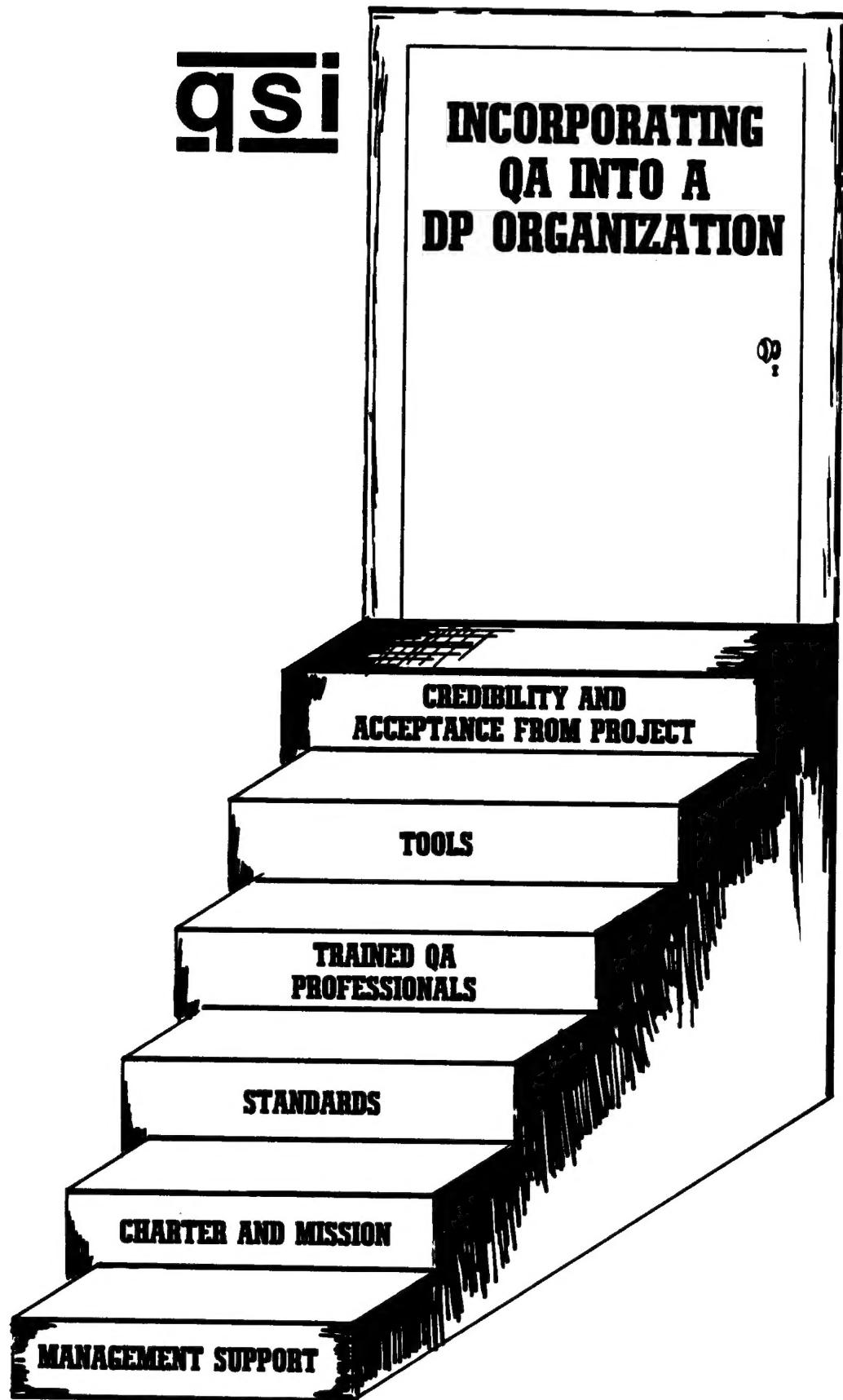
# **Contract Activities Overview**

## **Projects**

- Systems Integration
- Systems Development
- Systems Performance Evaluation and Enhancement
- Systems Quality Assurance

## **Method**

- Apply System Engineering Principles to Quality Assurance Problems





# **“MANAGEMENT SUPPORT HAS BEEN WON”**

## **QSI EXPERIENCE CORROBORATES**

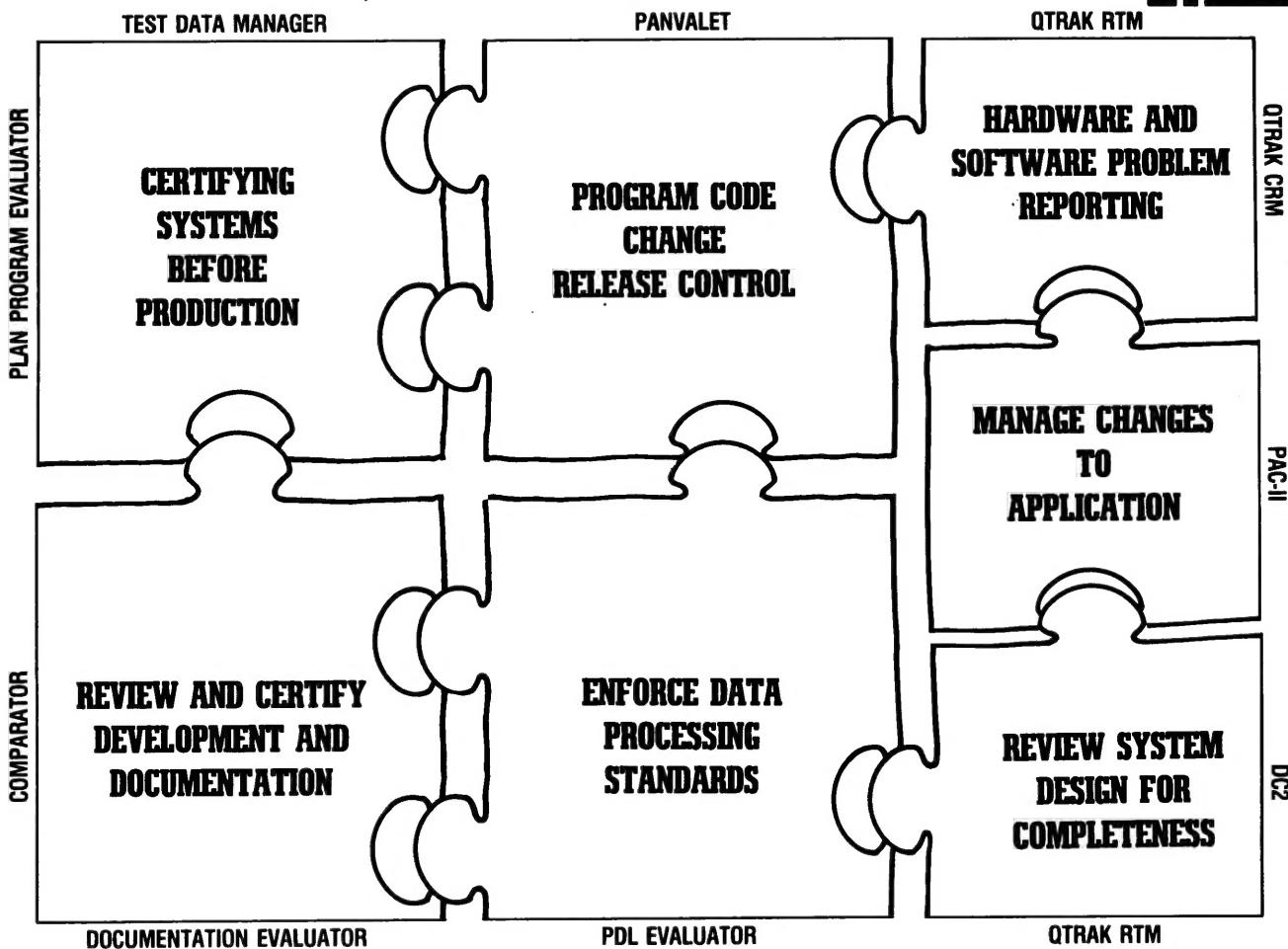
- QA Institute Survey
- Bill Perry's article in *Government Computer News*

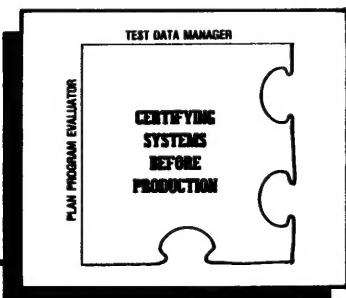
## **IN ADDITION**

- Charter and Mission Have Been Established
- Standards Are Being Used

# QA MISSION FUNCTIONS

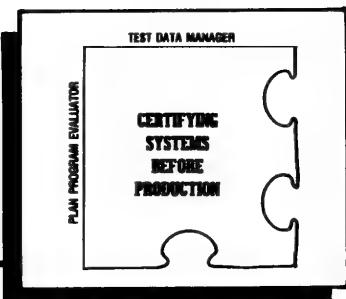
**qsi**





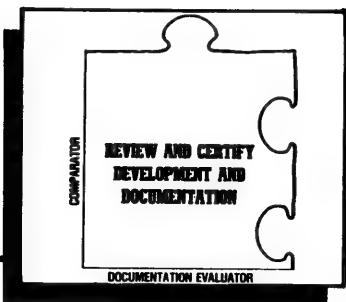
**TOOL NAME—Test Data Manager**

<b>WHAT IT IS</b>	<ul style="list-style-type: none"><li>• Automated test data control and generation program</li></ul>
<b>HOW IT WORKS</b>	Using structured specifications: <ul style="list-style-type: none"><li>• Generates test data for acceptance test</li><li>• Provides control of test data</li></ul>
<b>BENEFITS/ COSTS</b>	<ul style="list-style-type: none"><li>• Repeatability of test data assured</li><li>• Minimum resources required</li></ul>
<b>PROBLEMS</b>	<ul style="list-style-type: none"><li>• Start-up costs in small projects</li><li>• Large data storage requirements</li></ul>
<b>FUTURE</b>	<ul style="list-style-type: none"><li>• Reductions in storage media cost make data generator more and more attractive</li></ul>



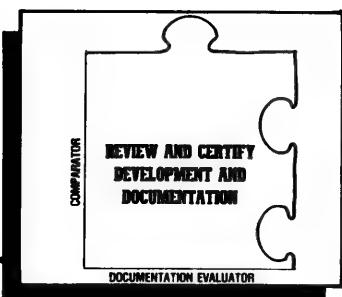
**TOOL NAME—Boole & Babbage Plan Program Evaluator**

<b>WHAT IT IS</b>	<ul style="list-style-type: none"><li>• Comprehensive extractions and reporting system to analyze system and problem performance</li></ul>
<b>HOW IT WORKS</b>	<ul style="list-style-type: none"><li>• Builds a comprehensive central file of data</li><li>• Provides activity usage profiles for evaluation</li></ul>
<b>BENEFITS/ COSTS</b>	<ul style="list-style-type: none"><li>• Time/usage profiles allow identification of small areas (5—10%) of code that use large (50—70%) amounts of resources</li><li>• Minimal code changes provide maximum time savings</li></ul>
<b>PROBLEMS</b>	<ul style="list-style-type: none"><li>• Large extractor data files</li><li>• Additional module analysis may reveal recoding requirements</li><li>• System priority required for fixed resource allocation</li></ul>
<b>FUTURE</b>	<ul style="list-style-type: none"><li>• Improvements in evaluator being made to support real-time examinations</li></ul>



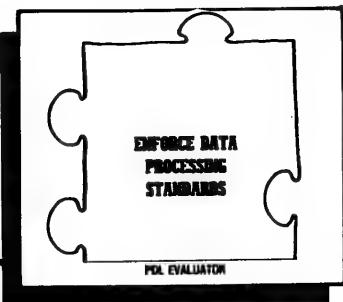
**TOOL NAME—Documentation Evaluator**

<b>WHAT IT IS</b>	<ul style="list-style-type: none"><li>• Program takes structured specification and compares it to standard</li></ul>
<b>HOW IT WORKS</b>	<ul style="list-style-type: none"><li>• Evaluates various levels of standard outline against document outline</li></ul>
<b>BENEFITS/ COSTS</b>	<ul style="list-style-type: none"><li>• First-line check on great volume of documentation</li><li>• Efficient and saves resources</li><li>• Simple program</li></ul>
<b>PROBLEMS</b>	<ul style="list-style-type: none"><li>• Requires maintenance</li><li>• Document and standard must be softcopy</li><li>• No content analysis and document specific</li></ul>
<b>FUTURE</b>	<ul style="list-style-type: none"><li>• Greater use on large projects but needs refining</li></ul>



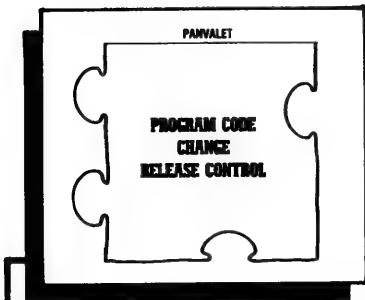
**TOOL NAME—Comparator**

<b>WHAT IT IS</b>	<ul style="list-style-type: none"><li>• Software library control program to compare two versions of the same computer program</li></ul>
<b>HOW IT WORKS</b>	<ul style="list-style-type: none"><li>• Reads two files and lists the differences</li></ul>
<b>BENEFITS/ COSTS</b>	<ul style="list-style-type: none"><li>• Identifies changes</li><li>• Simple to use</li><li>• Very efficient for large complex systems</li></ul>
<b>PROBLEMS</b>	<ul style="list-style-type: none"><li>• Initial reference point for each module must be explicitly defined</li></ul>
<b>FUTURE</b>	<ul style="list-style-type: none"><li>• Greater use on large projects</li></ul>



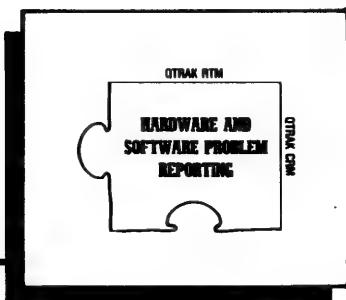
**TOOL NAME—PDL Evaluator**

<b>WHAT IT IS</b>	<ul style="list-style-type: none"><li>• Compliance checker for PDL constructs</li></ul>
<b>HOW IT WORKS</b>	<ul style="list-style-type: none"><li>• Checks constructs if-then-else, do-while, etc.</li><li>• Lists variations from pre-defined constructs</li></ul>
<b>BENEFITS/ COSTS</b>	<ul style="list-style-type: none"><li>• Ensures standards compliance</li><li>• Minimizes coding errors</li><li>• Ensures correct input to code generator</li></ul>
<b>PROBLEMS</b>	<ul style="list-style-type: none"><li>• Not a direct map from evaluation package to original standards</li></ul>
<b>FUTURE</b>	<ul style="list-style-type: none"><li>• Increased use to ensure more efficient and maintainable code</li></ul>



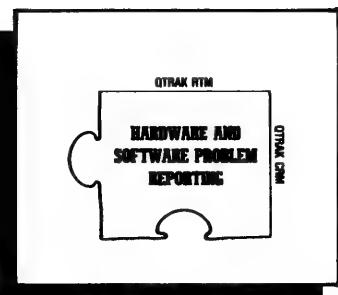
**TOOL NAME—Panvalet**

<b>WHAT IT IS</b>	<ul style="list-style-type: none"><li>• Program management and security system</li></ul>
<b>HOW IT WORKS</b>	<ul style="list-style-type: none"><li>• Builds and maintains a library of software modules</li><li>• Source code, object code, load modules, JCL</li><li>• Special text files</li></ul>
<b>BENEFITS/ COSTS</b>	<ul style="list-style-type: none"><li>• Flexible</li><li>• Efficient use of space</li><li>• Simple to use</li><li>• Compatible with most systems</li><li>• Good reporting functions</li></ul>
<b>PROBLEMS</b>	<ul style="list-style-type: none"><li>• Dedicated librarian required</li><li>• User training necessary</li></ul>
<b>FUTURE</b>	<ul style="list-style-type: none"><li>• Batch system use extended to VM/CMS environment</li><li>• Growing use of library management and control systems</li></ul>



**TOOL NAME—Automatic Change Control, Management & Distribution System (CMD)**

<b>WHAT IT IS</b>	<ul style="list-style-type: none"><li>• Information management system for electronic mail, file management, and user communication</li></ul>
<b>HOW IT WORKS</b>	<ul style="list-style-type: none"><li>• Allows users to create format, review, send, and receive documents in a controlled, easy manner</li></ul>
<b>BENEFITS/ COSTS</b>	<ul style="list-style-type: none"><li>• Solves physical space storage problems</li><li>• Access is controlled by classification, passwords, and privileges</li><li>• Extensive accessibility</li></ul>
<b>PROBLEMS</b>	<ul style="list-style-type: none"><li>• Electronic mail is dependent on a reliable and available computer system</li><li>• No backup</li><li>• System administrator required</li></ul>
<b>FUTURE</b>	<ul style="list-style-type: none"><li>• Continuing improvement in user language and human engineering factors being made to enhance system</li></ul>



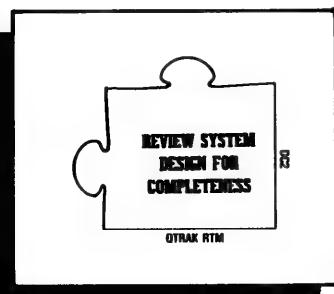
**TOOL NAME—QTRAK Communications & Reporting Module (CRM)**

<b>WHAT IT IS</b>	<ul style="list-style-type: none"><li>• Tracking system for problem reporting</li></ul>
<b>HOW IT WORKS</b>	<ul style="list-style-type: none"><li>• Data base query and reporting module</li><li>• Audit, track, report on problem data base</li></ul>
<b>BENEFITS/ COSTS</b>	<ul style="list-style-type: none"><li>• Minimizes data entry</li><li>• Real-time status reports</li><li>• Programmers can query CM records from terminal</li></ul>
<b>PROBLEMS</b>	<p>Requires:</p> <ul style="list-style-type: none"><li>• Training</li><li>• Maintenance</li><li>• Dedicated system administrator</li></ul>
<b>FUTURE</b>	<ul style="list-style-type: none"><li>• Increased use, most current status accounting reports possible</li></ul>



**TOOL NAME—PAC-II**

<b>WHAT IT IS</b>	<ul style="list-style-type: none"><li>• Computerized project management system</li></ul>
<b>HOW IT WORKS</b>	<ul style="list-style-type: none"><li>• Allows planning, budgeting, monitoring, analyzing, and costing of projects</li><li>• Batch/interactive system</li></ul>
<b>BENEFITS/ COSTS</b>	<ul style="list-style-type: none"><li>• Planning analysis using interactive system</li><li>• Plethora of management reports</li><li>• Graphic capabilities</li></ul>
<b>PROBLEMS</b>	<ul style="list-style-type: none"><li>• Start-up time required</li><li>• Planner/administrator training required</li></ul>
<b>FUTURE</b>	<ul style="list-style-type: none"><li>• Improvements being made in MIS graphics interfaces</li><li>• Expanded use of exception reporting features</li></ul>



**TOOL NAME—QTRAK Requirements Module (RTM)**

<b>WHAT IT IS</b>	<ul style="list-style-type: none"><li>• Automated requirements traceability matrix</li></ul>
<b>HOW IT WORKS</b>	<ul style="list-style-type: none"><li>• Module for tracking requirements</li><li>• Indexes on requirements and documents</li></ul>
<b>BENEFITS/ COSTS</b>	<ul style="list-style-type: none"><li>• Requirements traced through life cycle</li><li>• Automated exception notification</li><li>• Quick, easy to use</li></ul>
<b>PROBLEMS</b>	<ul style="list-style-type: none"><li>• Requires maintenance</li><li>• Data integrity must be assured</li></ul>
<b>FUTURE</b>	<ul style="list-style-type: none"><li>• Increased use as systems become more complex</li></ul>

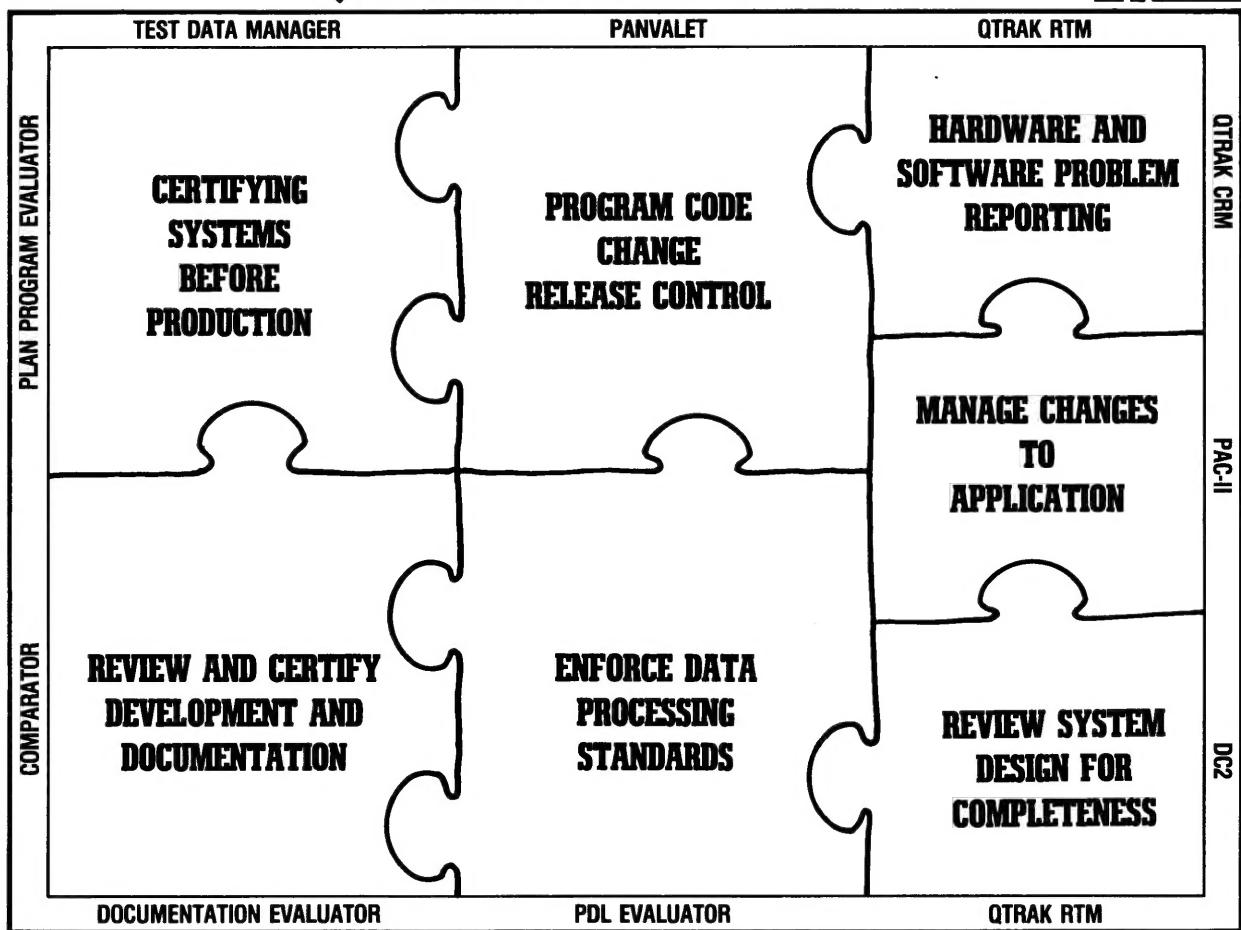


#### TOOL NAME—Data Catalogue 2

<b>WHAT IT IS</b>	<ul style="list-style-type: none"><li>• Automated data dictionary and tracking system</li></ul>
<b>HOW IT WORKS</b>	<ul style="list-style-type: none"><li>• Establishes relational and hierachial</li><li>• Automatically track, monitor, and query to data element level</li></ul>
<b>BENEFITS/ COSTS</b>	<ul style="list-style-type: none"><li>• Automatic generation of program structures COBOL, PL-1, ASSEMBLER</li><li>• Interactive query and update features</li><li>• On-line report generators</li></ul>
<b>PROBLEMS</b>	<ul style="list-style-type: none"><li>• Complex system to be used with large problems</li><li>• Startup time</li><li>• User training</li></ul>
<b>FUTURE</b>	<ul style="list-style-type: none"><li>• System improvements being made for custom dictionary facility to be used with top-down structured methods</li></ul>

# QA MISSION FUNCTIONS

qsi





# **TODAY'S QA ISSUES**

## **PROBLEMS**

- Significant gap in credibility and support from project personnel
  - Lack of trained QA professionals
  - Insufficient use of tools to support QA mission functions

## **SOLUTIONS**

- Increased professional training
- Tools as a solution
  - Help projects do a better job
  - Find potential problems before become problems
  - Provide impartial judgment
  - Use most current methods



# **SUMMARY**

## **FIRST STEPS HAVE BEEN MADE FOR INCORPORATION**

- Management Support
- Charter and Mission
- Standards

## **THE NEXT STEPS**

- Winning Credibility and Acceptance
  - Training QA Professionals
  - Greater Use of Tools

## **QA MUST**

- Match the Available Tools with Functions
- Optimize Use
- Demonstrate How Tools Can Make QA and Workers More Efficient
- Develop Additional Tools